Introducing PS2 to PC Programmers

David Carter
SCEE Technology Group

Contact

- Contact Information:
  - SCEE Booth Exhibition Stand #9

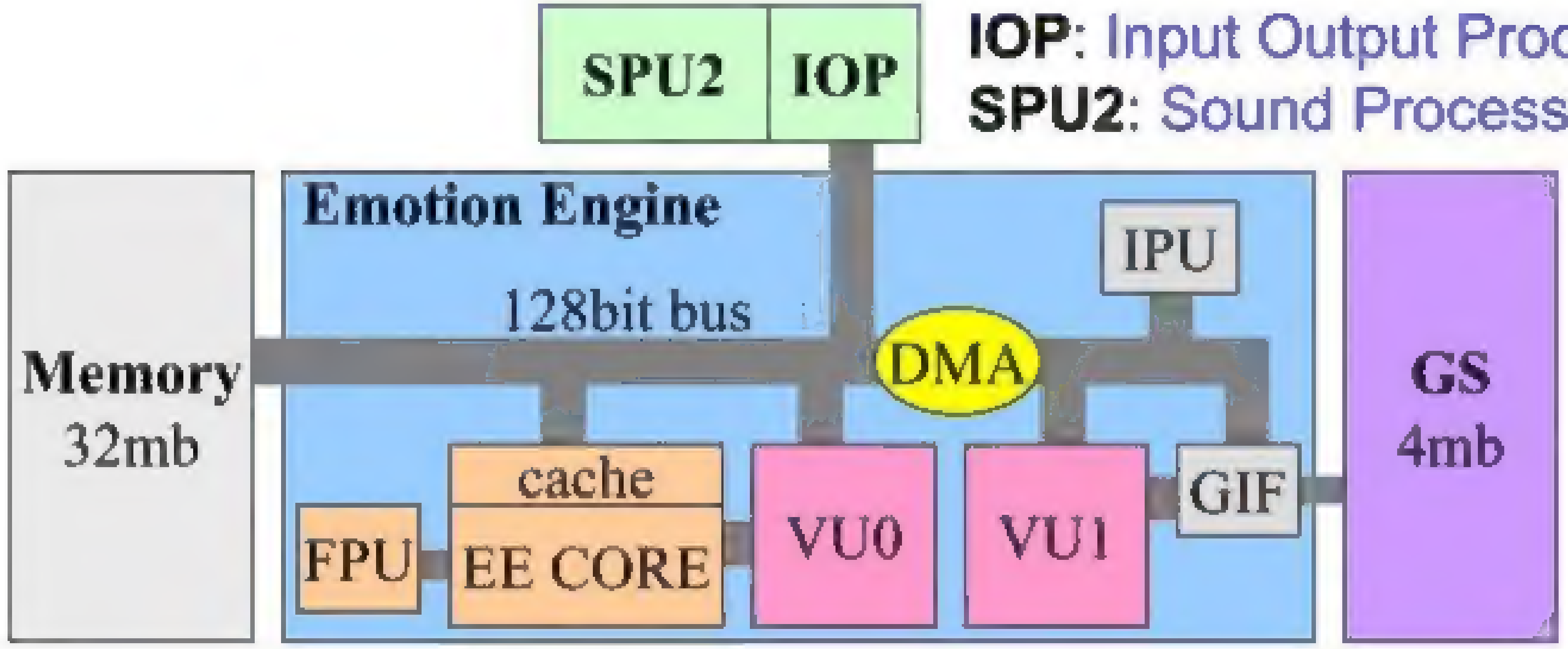
What We Will Be Covering

- An overview of the hardware
- A basic rendering pipeline
- How to improve performance
- Under used capacities
- PS2 design techniques
- Questions…

What We Will Not Be Covering

- A MIPS programming course
- Showing any sample code
- The price of beer (I am so glad it is cheap!)
- A PS2 in chocolate (ummm…tasty!)

Basic PS2 Architecture

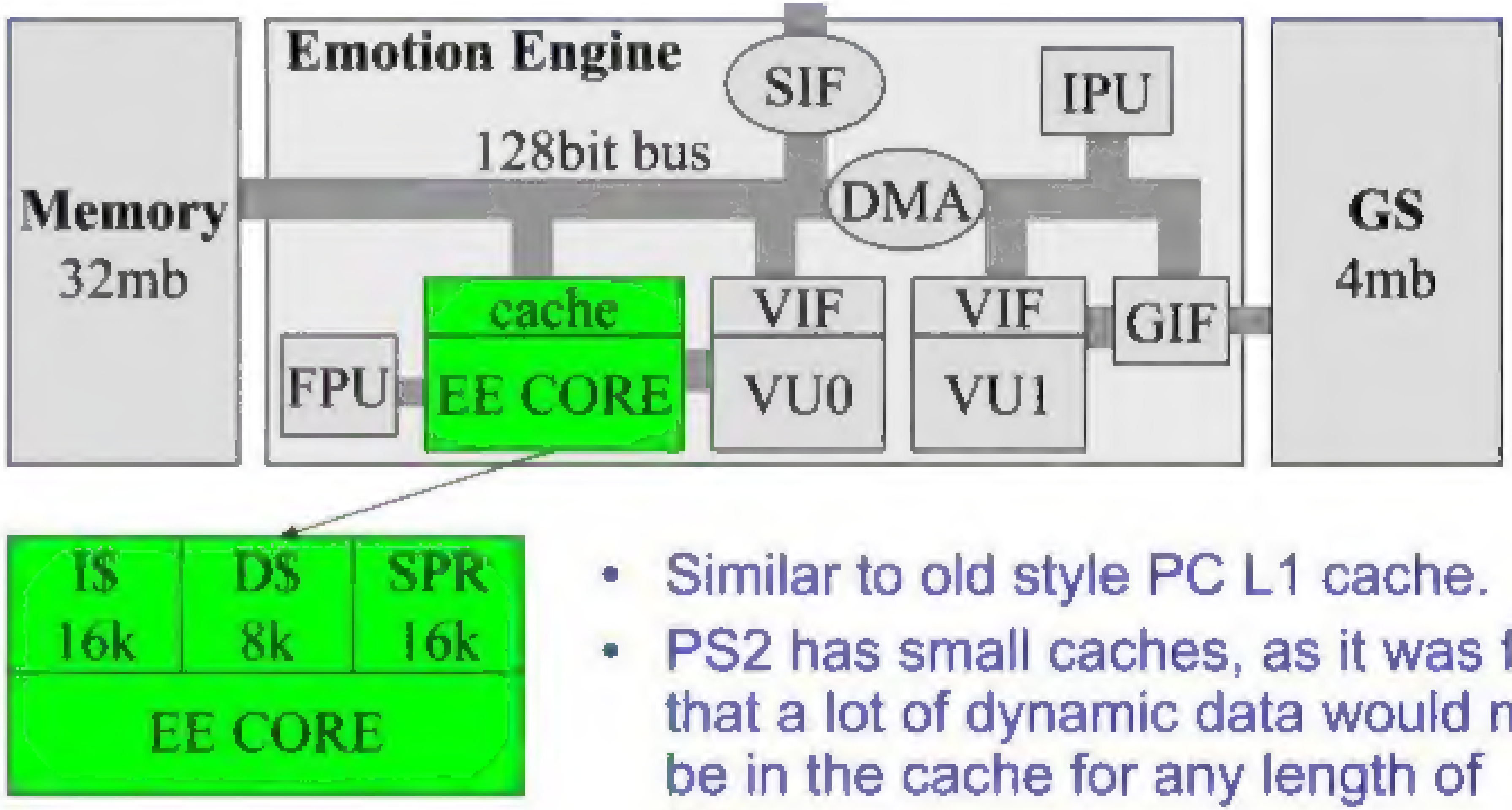


IOP: Input Output Processor
SPU2: Sound Processor

EE: 128-bit Emotion Engine
VU0/VU1: Vector Units
FPU: Floating Point Unit

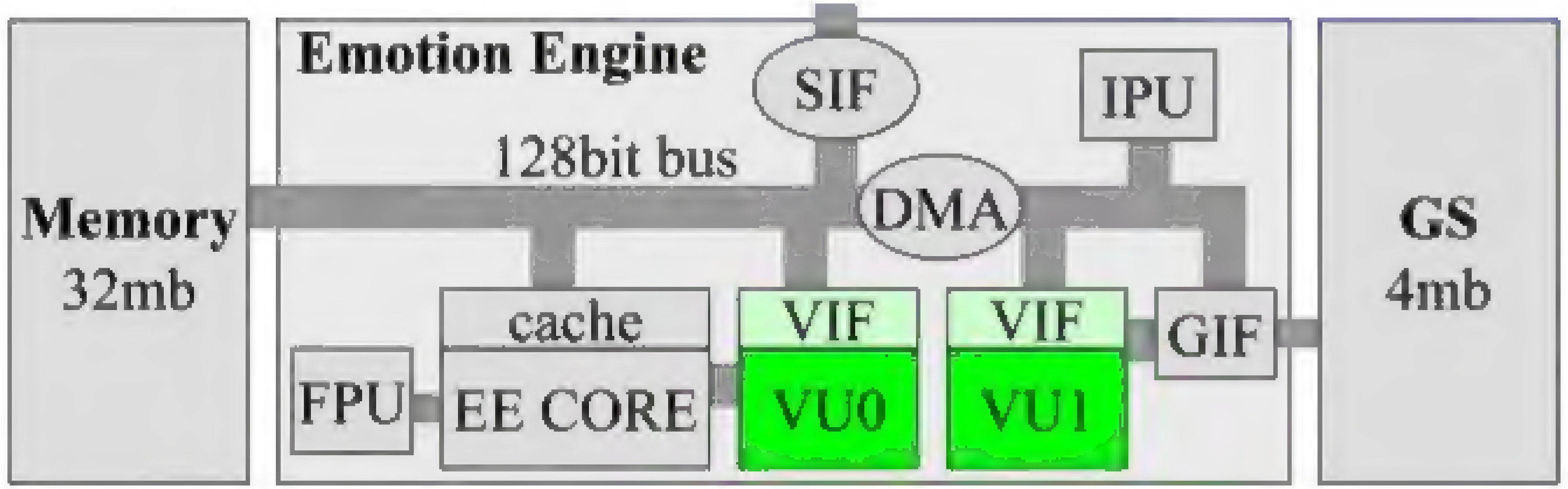
GS: Graphic Synthesiser
DMA: Direct memory access
IPU: Image processing Unit

Caches And Scratchpad



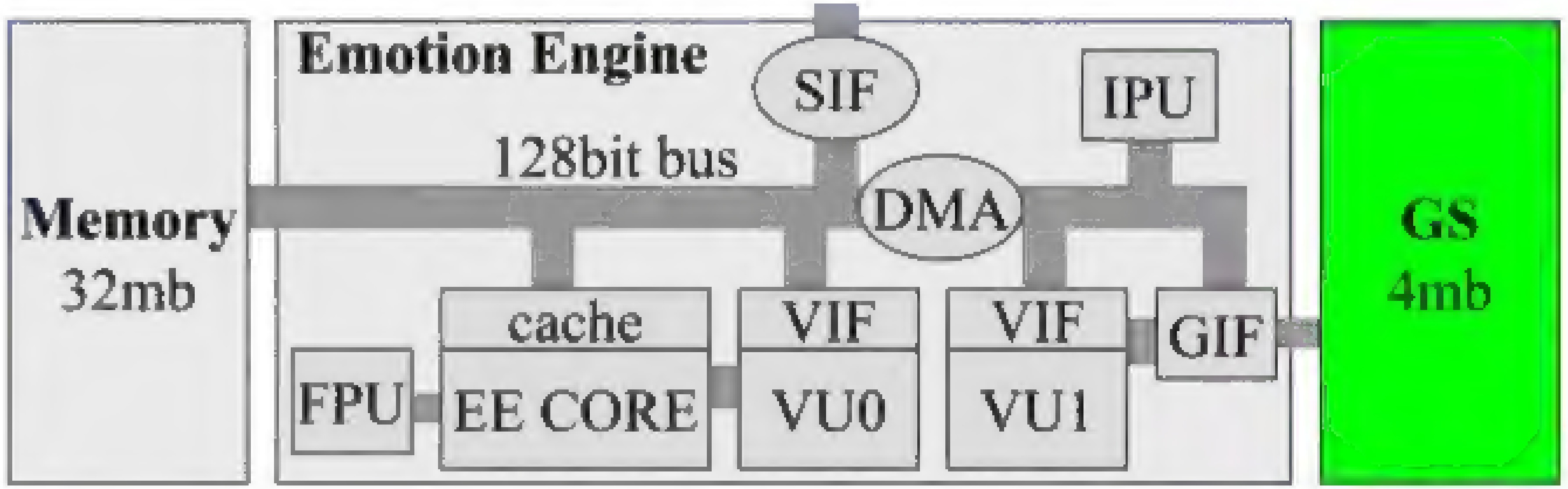
- Similar to old style PC L1 cache.
- PS2 has small caches, as it was felt that a lot of dynamic data would not be in the cache for any length of time.

EE Vector Units



- Each vector unit can do 4 multiplies and 4 adds in a single instruction and can transform about 36million vertices/sec.
- Both can operate in Micromode – LIW architecture (32bits*2)
- Argued that due to the PS2 architecture the PC paradigm started to shift with the emergence of Vertex Shaders.

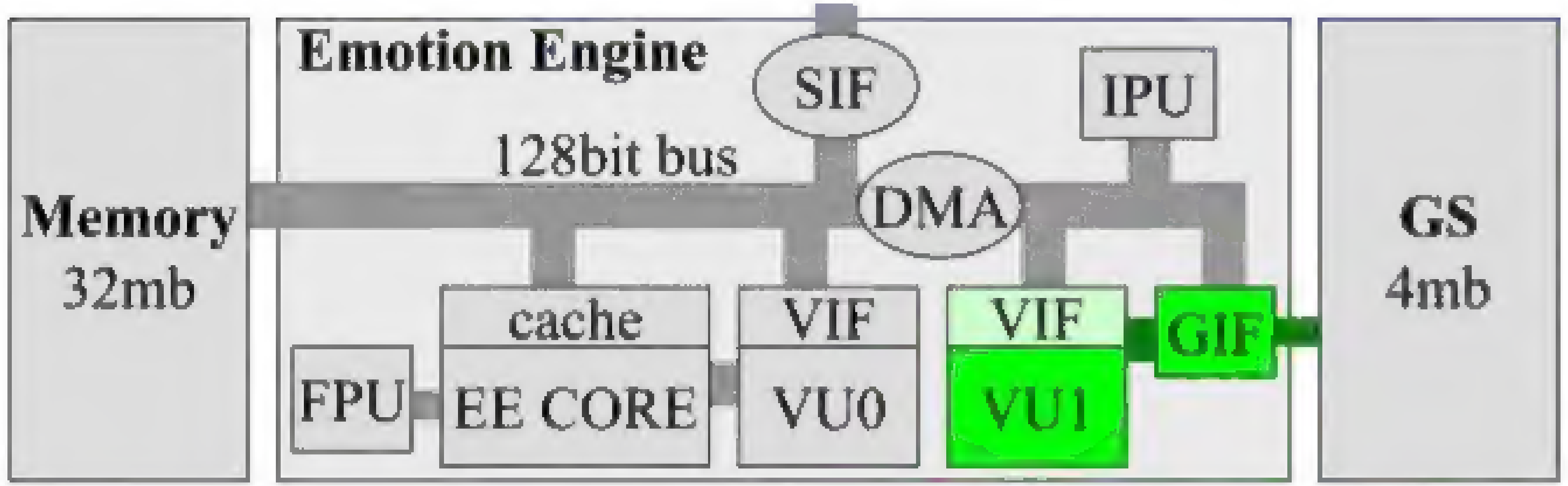
Graphic Synthesiser



Primitives per second:
150million points
50million textured sprites
75million untextured triangles
37.5million textured triangles

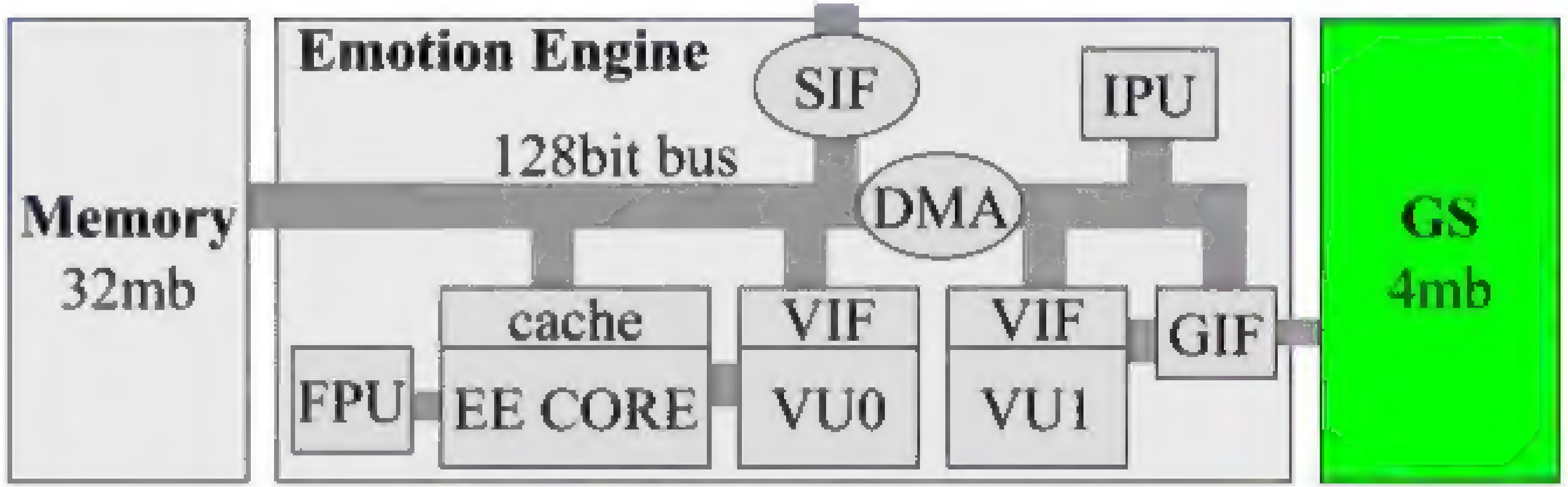
Features:
Alpha blend, Z-test, Bi-linear/tri-linear filtering. Efficient scissoring and a fill rate of **2.4**-giga pixel.

GIF Connection For VU1



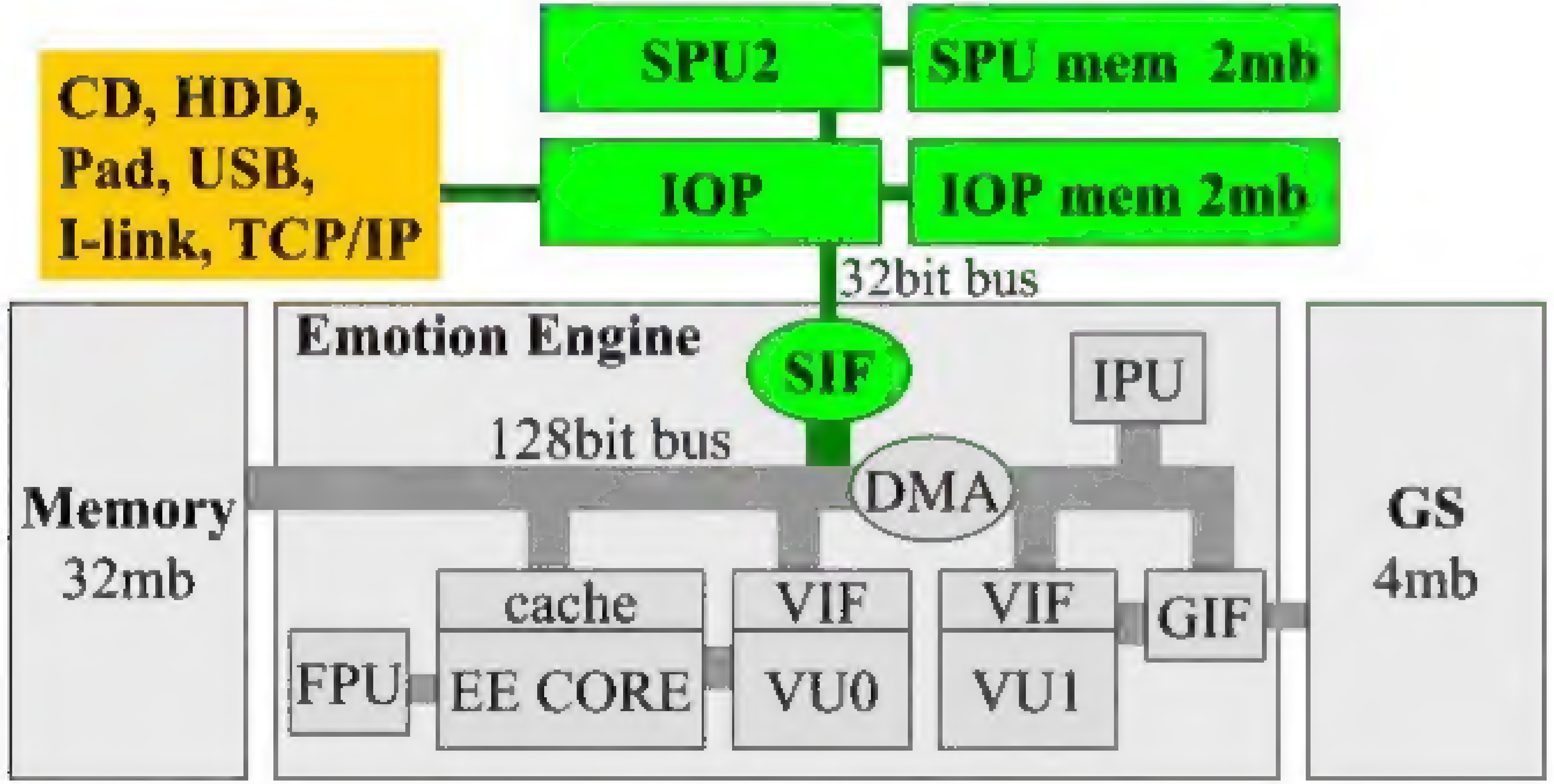
- Vector Unit 1 has a dedicated output path to the GIF
- It also has a much larger internal memory than VU0 to support double buffering of input and output data.
- This enables fast transformation and output to GS of patterned data.

Fill Rate



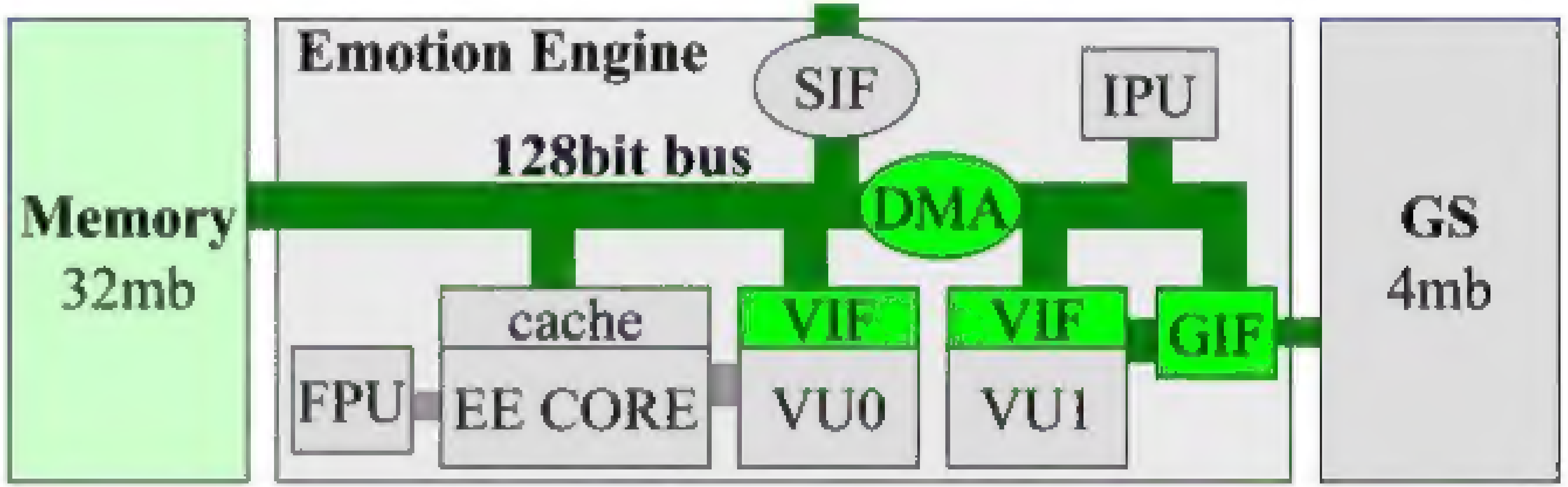
- Bandwidth of 4MB Embedded DRAM 48GB/sec
  - Bandwidth of frame buffer 38.4Gb/sec
  - Texture bandwidth 9.6Gb/sec
- Fill rate 1.2Giga pixel a sec for texture
- Fill rate 2.4Giga pixel a sec for untextured

IOP, SPU And Backwards Compatibility



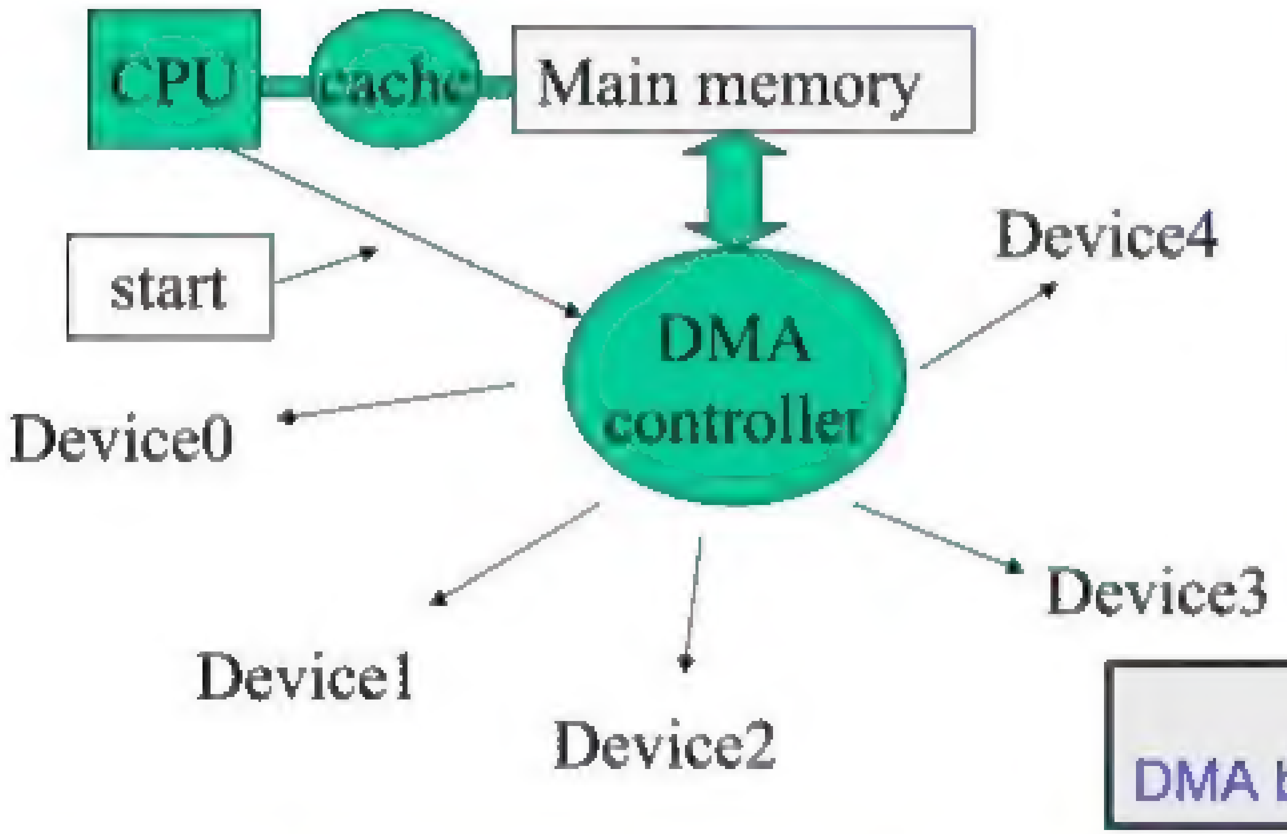
The IOP processor comes from PS1, this solves compatibility!

DMA



- DMA bus has a bandwidth of 2.4Gb/sec, faster than AGPx8 which is (in theory!) 2.1Gb/sec.
- The DMA bus controls all data transfers in the system.
- The DMAC will not stall the CPU when transferring data.
- DMA transfers must be aligned to 128bits.

DMA Data Transfer



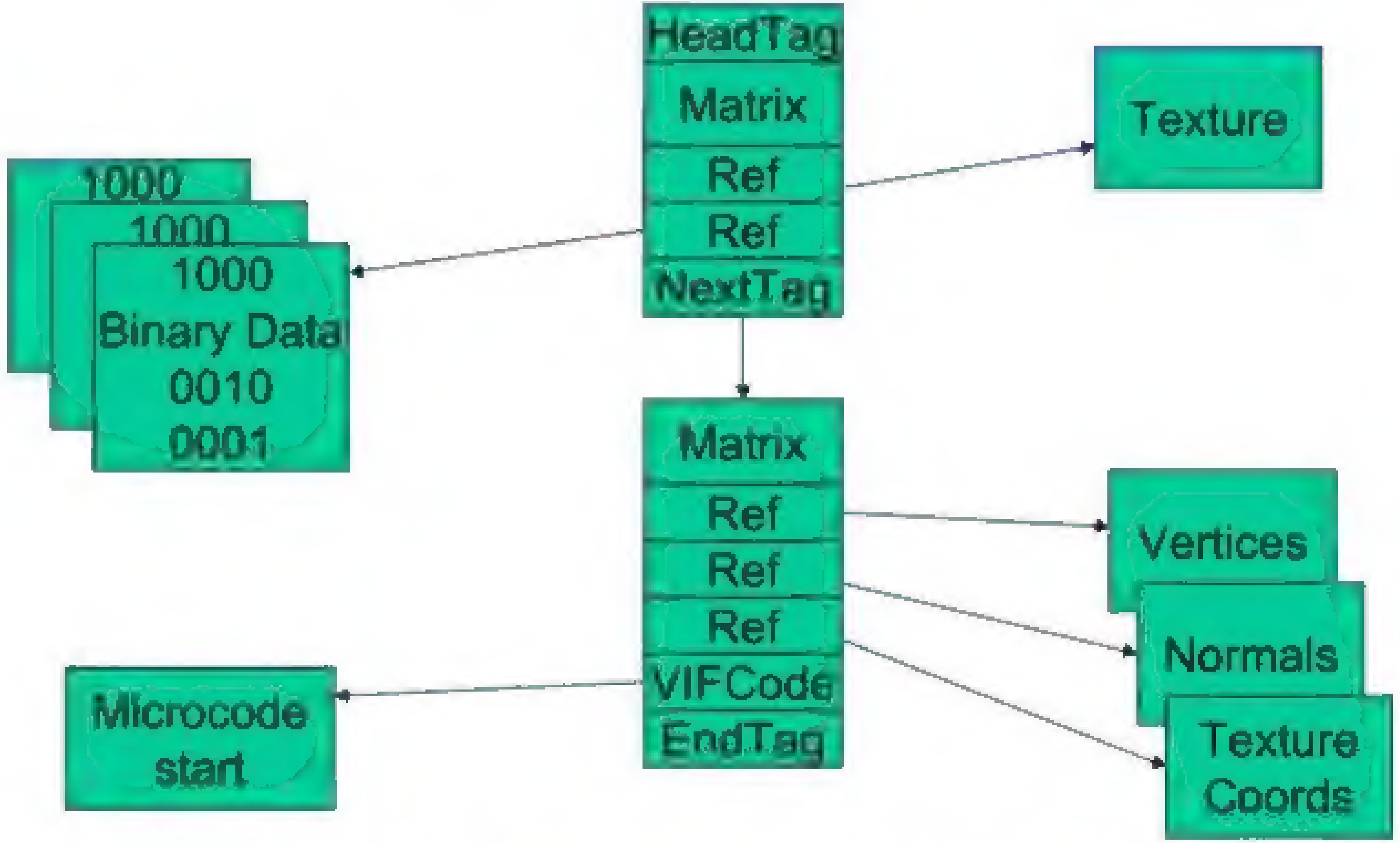
Time sliced:
8qword to 1
8qword to 2
8qword to 3
repeat

Dedicated channel for each device

NOTE
DMA bypasses the cache

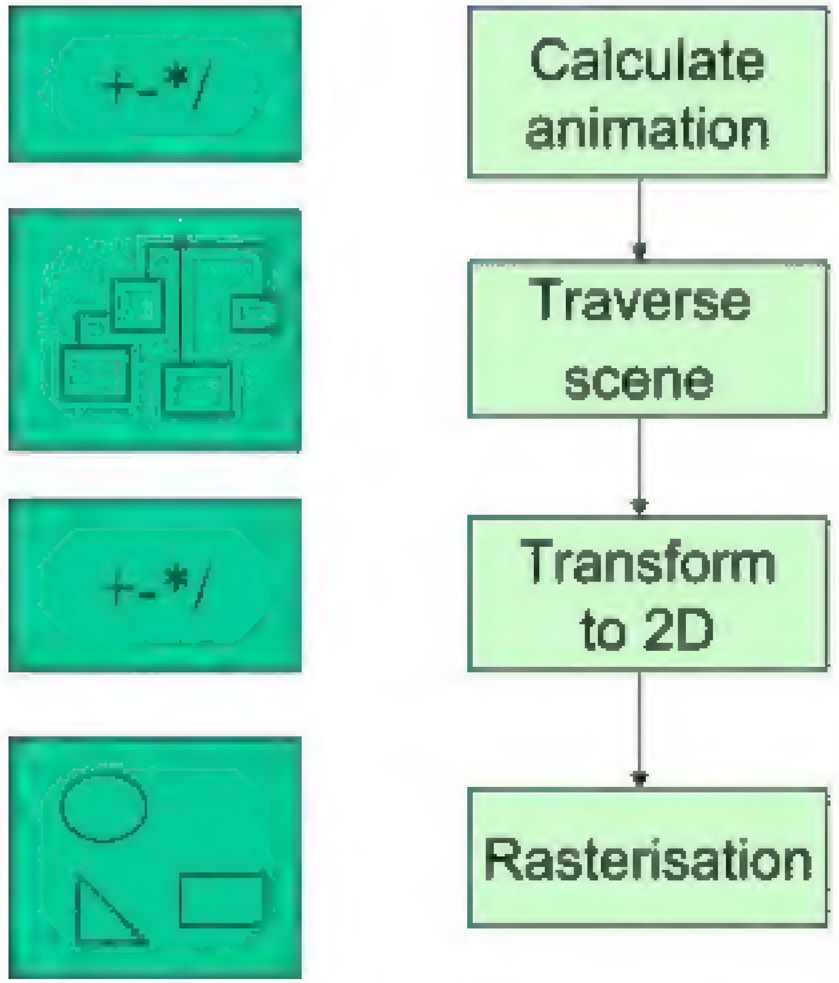
To send data through a channel you just specify the start address, the data size and a start signal to the DMAC.

DMA Chains



Built from list of tags, can contain many data types

Basic Rendering Pipeline



Calculate animation — CPU + coprocessor VU0

Traverse scene — List processing DMA

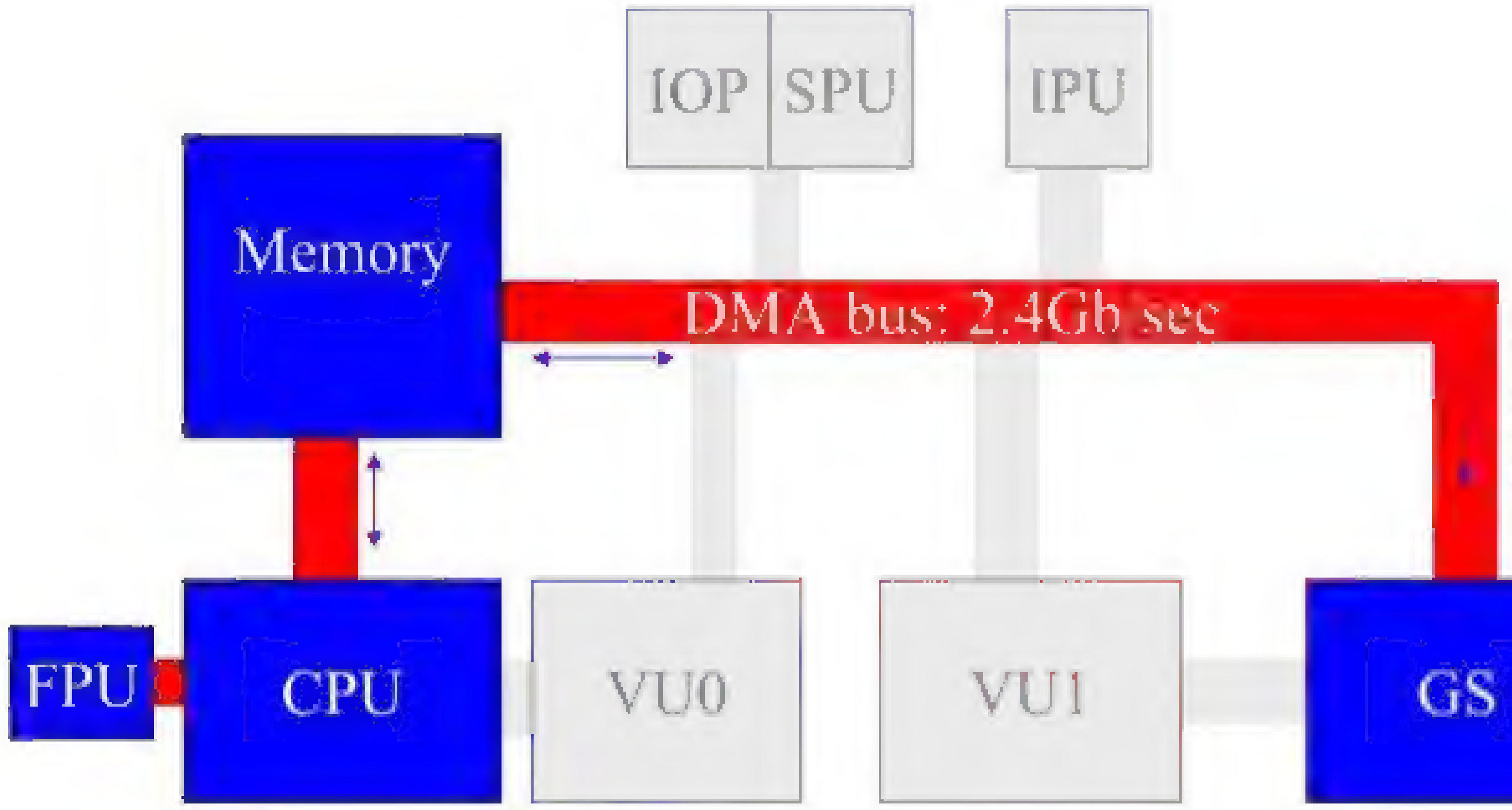
Transform to 2D — VU1

Rasterisation — GS

How To Improve PS2 Performance

- By not treating the PS2 as a PC
- By using texture sizes and formats
- Prevent the thrashing of Texture Cache
- Without abusing Instruction and Data Cache

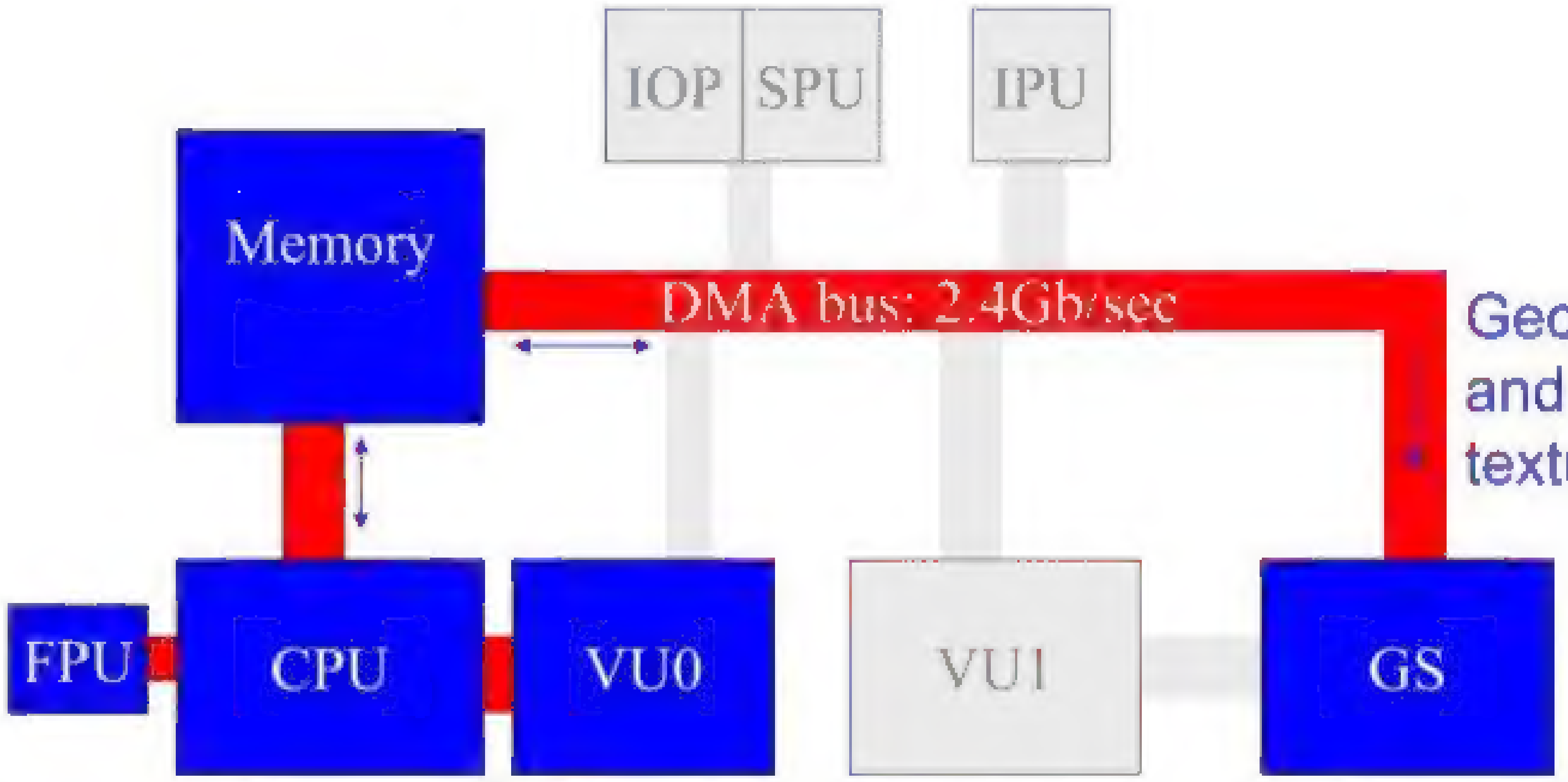
1st Attempt At A PC Port (max 0.5 million polys)



Geometry and texture

Transformation

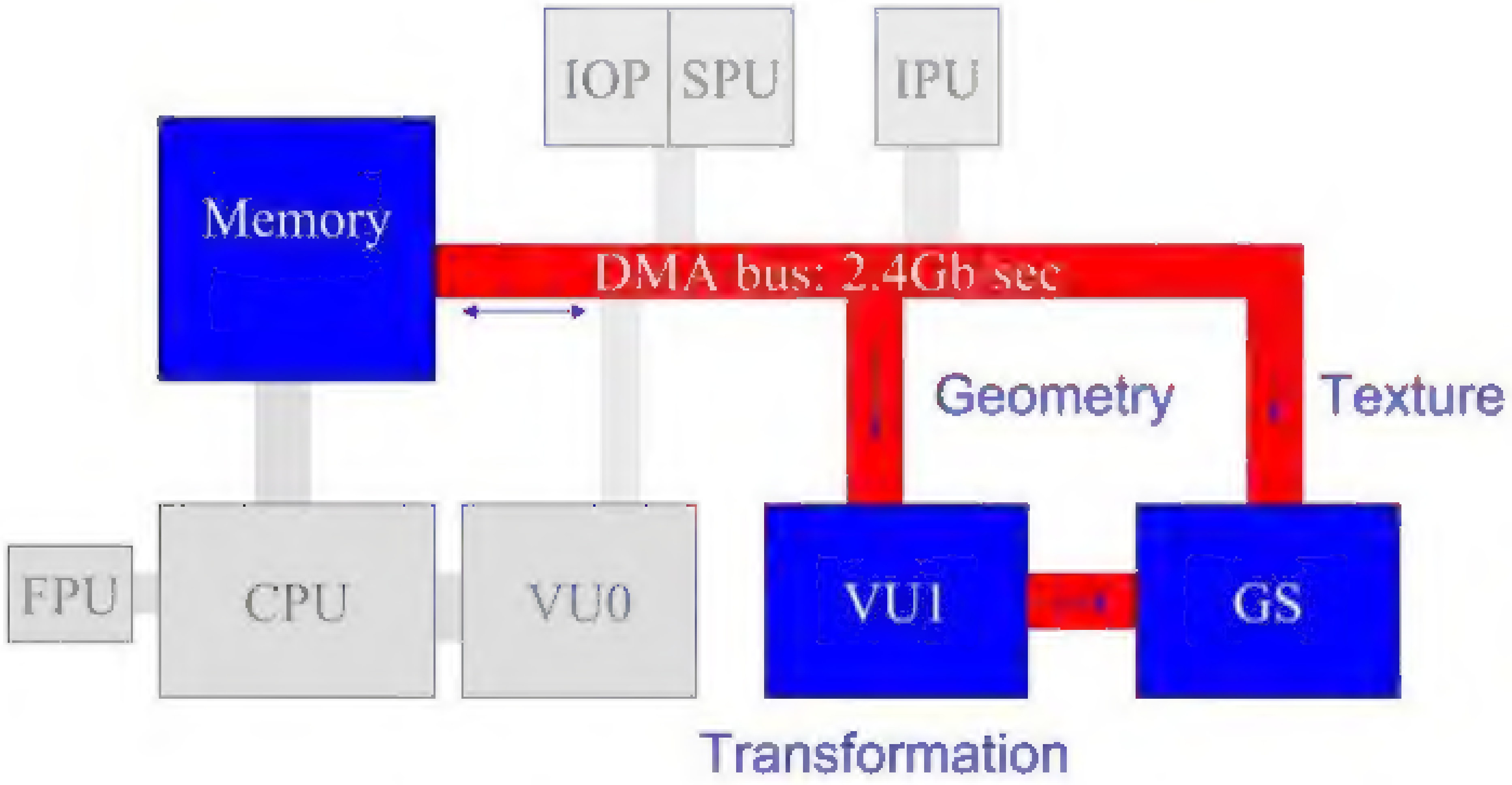
2nd Attempt At A PC Port (max 1.5 million polys)



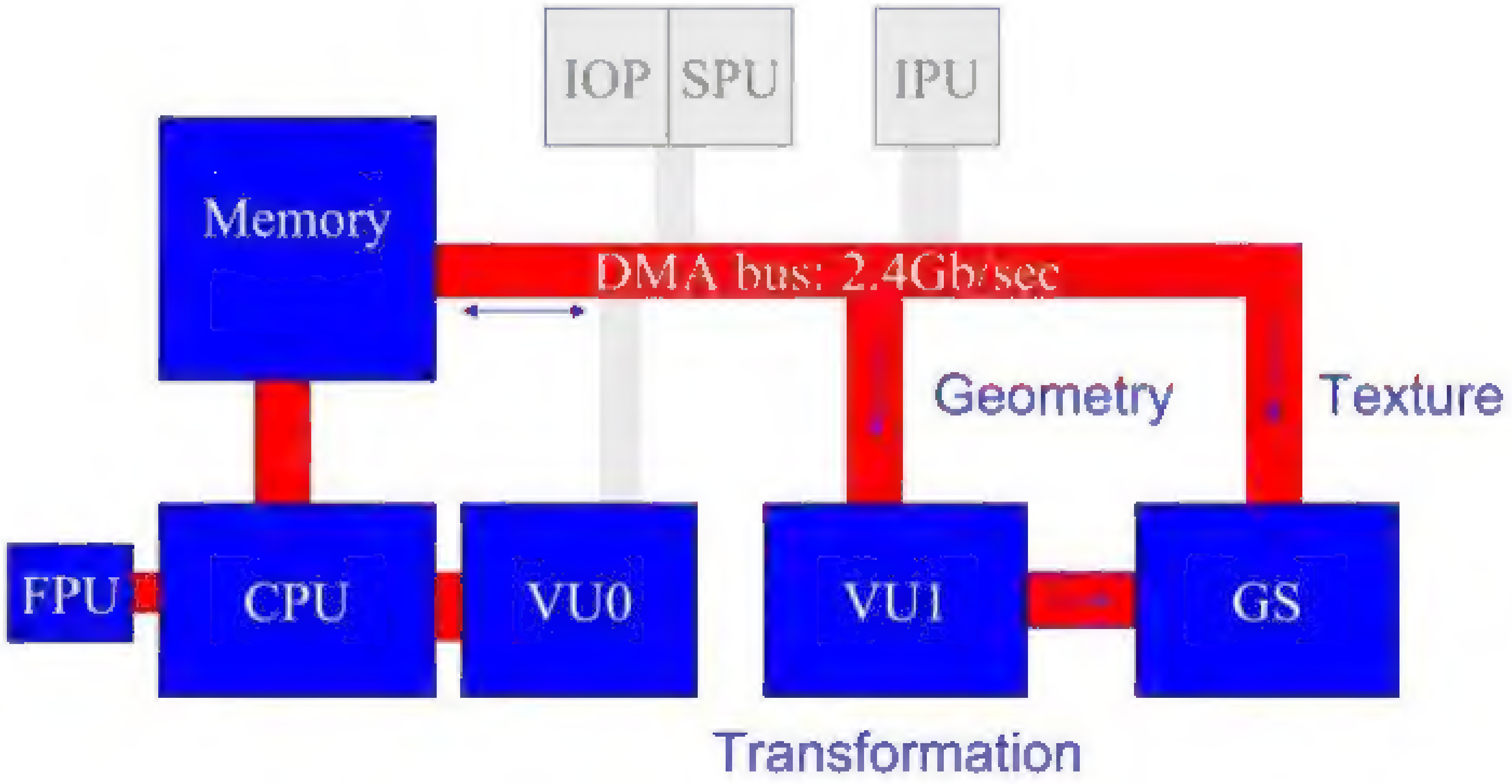
Geometry and texture

Transformation in parallel with CPU

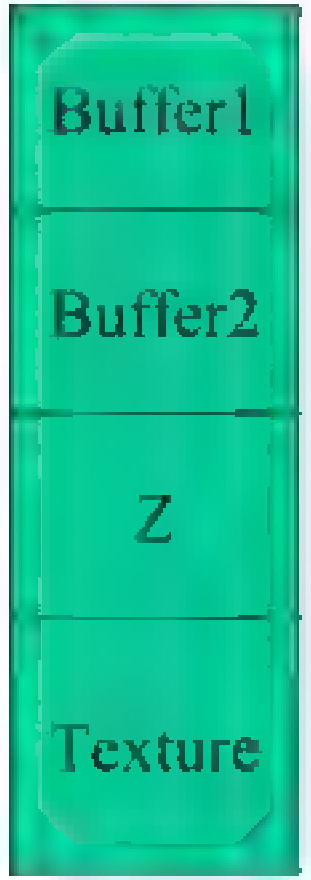
VU Renderer (lighting, no animation) (typical 10-20 million polys)



Complete Game (lighting, animation) (typical 5-10 million polys)



VRAM Layout

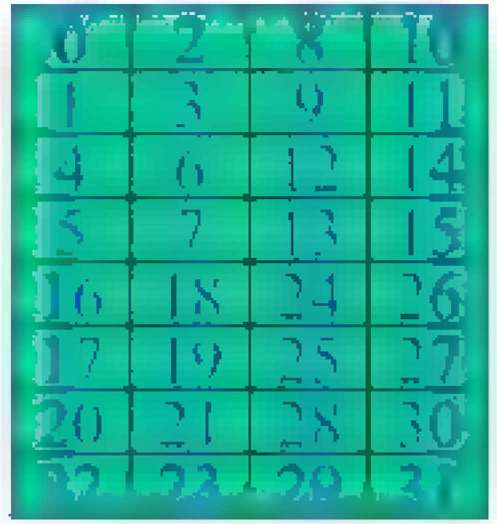


- 4MB Embedded memory
- 4MB of VRAM is split into 8K pages
  - Pages split into 32 blocks of 256 bytes
- Frame buffers addressed by page
- Textures addressed by block
  - Allowing multiple textures per page

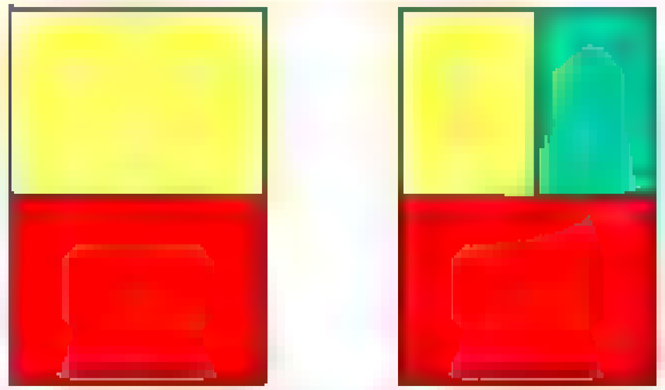
By Using Texture Size And Format

- 4MB of VRAM is split into 8K pages
  - Pages split into 32 blocks of 256 bytes
- Block position varies based on format
- Possible to store multiple textures in 1 page

- EG 16-Bit Texture Page



0	2	8	10
1	3	9	11
4	6	12	14
5	7	13	15
16	18	24	26
17	19	25	27
20	21	28	30
22	23	29	31

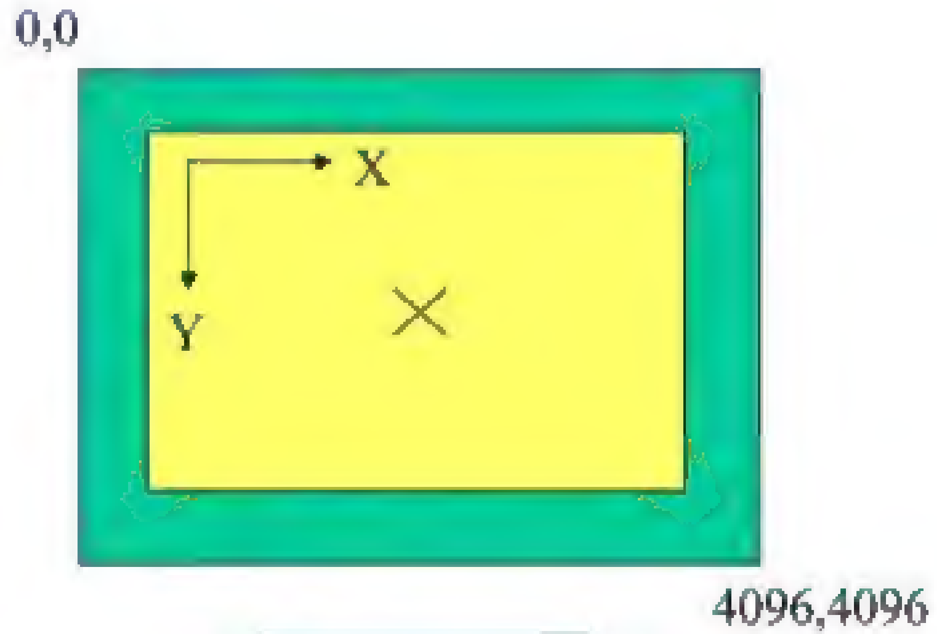


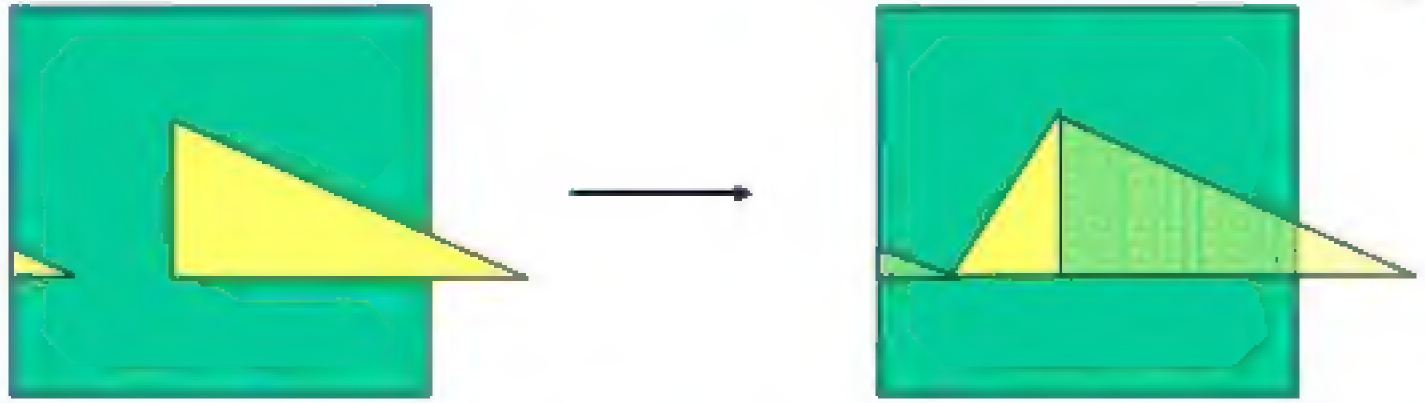
GS Coordinate System

- Frame Buffers use a 16-bit coordinate system
  - 12-bit integer . 4-bit fraction
  - Full Range 0 - 4095.9375
- Typically centre specified as (2048, 2048)
- Scissoring area specified based relative to this centre

GS Coordinate Scissoring

- X and Y Values are 16bit
  - Scissoring will not work outside that range
- No hardware clipping
  - There is a VU clip instruction





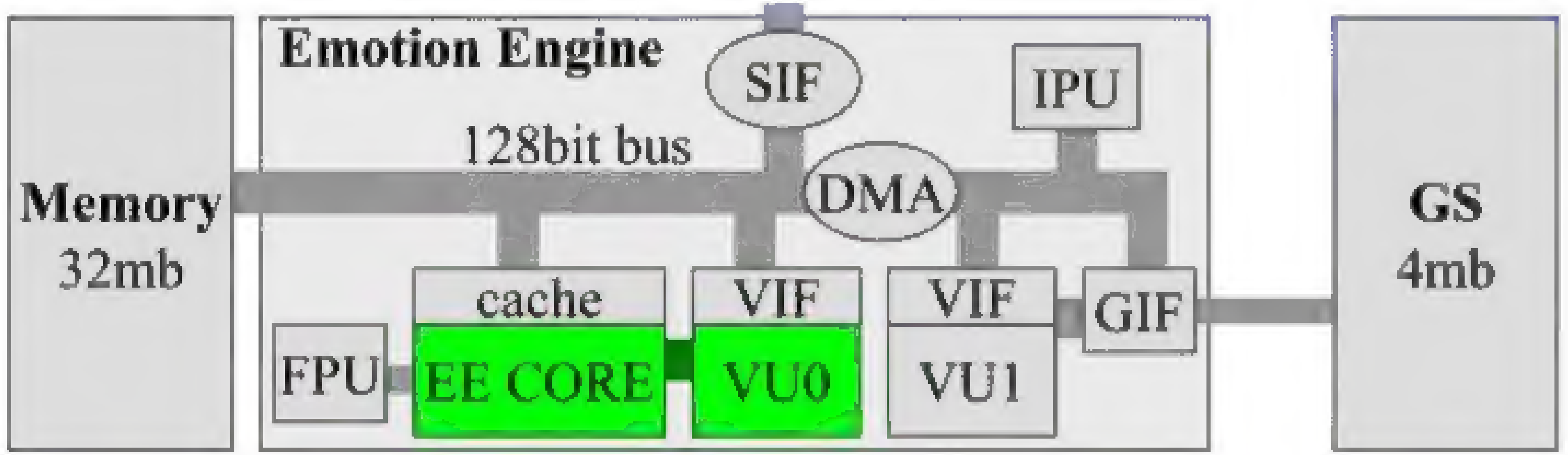
Prevent The Thrashing Of Texture Cache

- Current texels read from Texture Cache
  - Only 8K in size or 1 Texture Page
  - Costs to reload Texture Cache
- No need to use PC-style 32-bit textures
  - Too many colours, takes up too much VRAM
  - Aiming for TV not a PC Monitor
- Texture Sizes that fit into Texture Cache
  - 4bit 128x128, 8bit 128x64 (with CLUT)
  - 16bit 64x64, 32bit 64x32

Instruction And Data Cache Issues

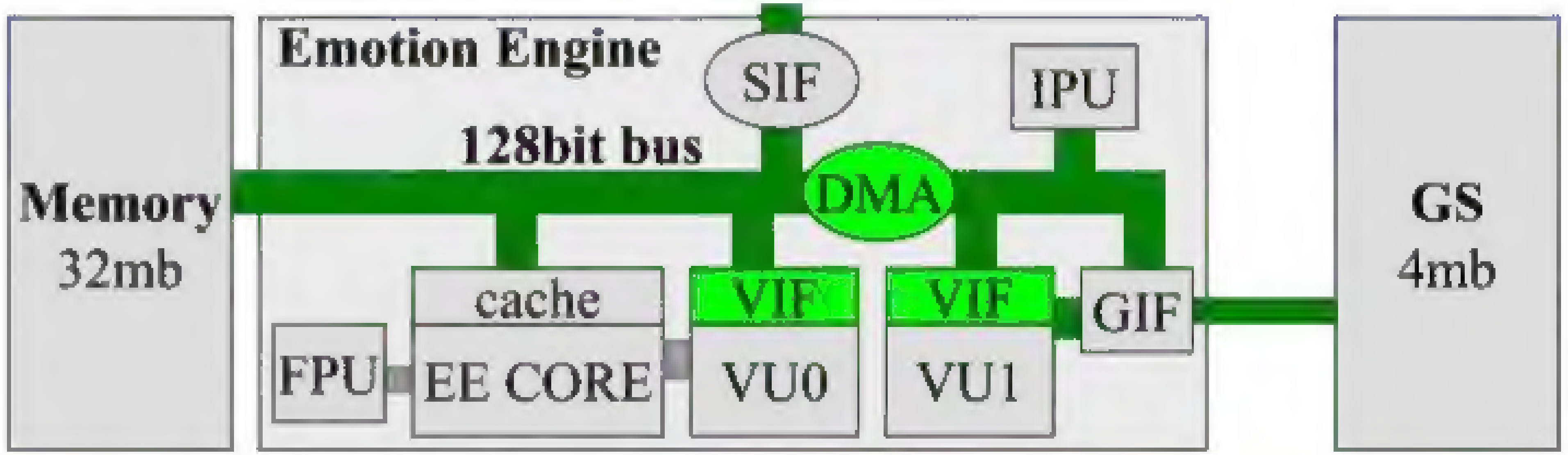
- Cache Issues
  - Large Loops and Jumps
  - Large Objects/Structures
  - Consider the cost of useful C++ features (e.g. Templates) they can have a negative effect
- What can help?
  - Breaking large loops into several smaller loops
  - Check disassembly of code for inlining
  - Un-cached Memory Access (0x20000000)
  - Scratchpad is the fastest memory you have direct access to, use as a main work area.

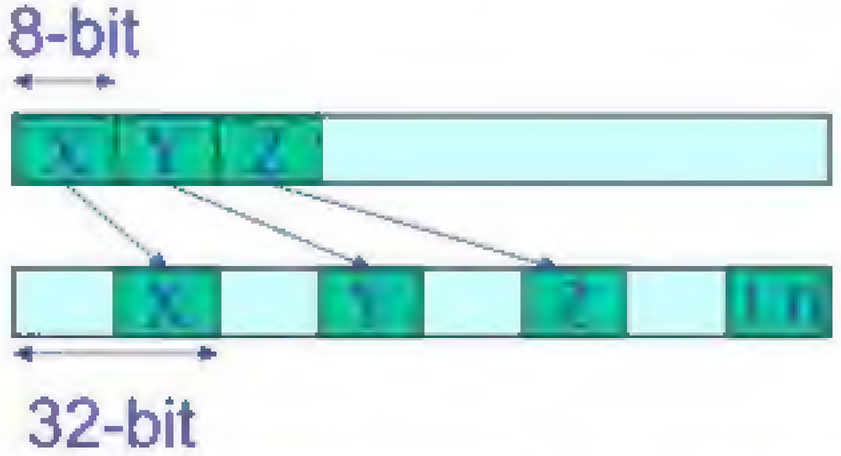
Vector Unit 0 Usage



- Suggested for taking some work off the CPU and help reduce I$ misses.
- Its not recommended to use VU0 in Macromode.
- Use Micromode and allow the CPU to carry on in parallel.

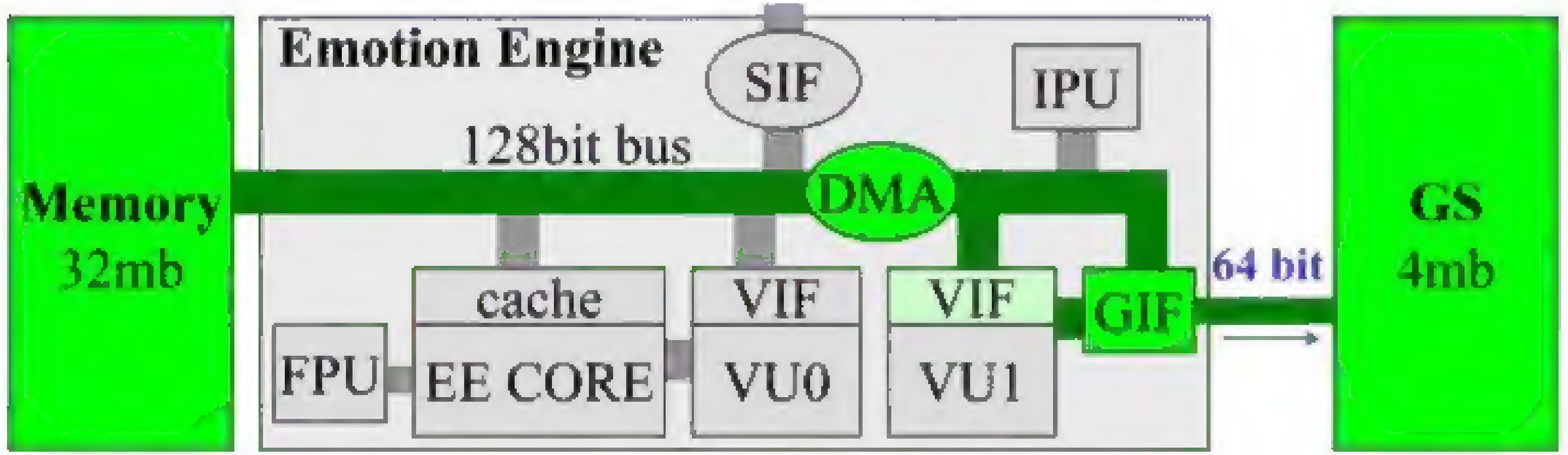
VIF Data Compression/Decompression





- Compressed formats reduce memory size of model.
- Decompression from packed formats by the VIF, provides reduction load on VU.

Texture And Geometry Streaming



- 1.2Gb/sec max bandwidth (24-meg/frame).
- GIF arbitrates between paths and packs data in to 64bit for GS.
- Watch priority ordering with paths to the GIF.

Summary

- The key to PS2 power is keeping the units busy
- Keeping data moving in parallel is the key to keeping the processors fed with data.
- DMA is the system which does this. This is the most crucial thing to understand to get performance on PS2.
- VRAM seems small but there are plenty of tricks.
- Cache issues… remember Scratchpad!
- Vector Unit 0 is underused.